



DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON

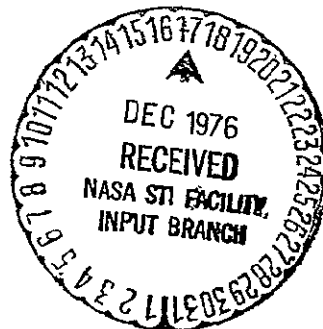
CR-151133
HOUSTON, TEXAS

(NASA-CR-151133) USER'S GUIDE: DATEXT
(Houston Univ.) 33 p HC A03/MF A01 CSCL 09B

N77-13696

Unclas
G3/61 56940

USER'S GUIDE: DATEXT
BY WILLIAM A. COBERLY
JACK D. TUBBS
LARRY HINMAN
REPORT #58 AUGUST 1976



PREPARED FOR
EARTH OBSERVATION DIVISION, JSC
UNDER
CONTRACT NAS-9-15000

HOUSTON, TEXAS 77004

User's Guide: DATEXT

by

*William A. Coberly, University of Tulsa, University of Houston
Jack D. Tubbs, NRC Postdoctoral Fellow-JSC/MPAD
Larry Hinman, Aeronutronic Ford, University of Houston*

[OS/360 Dependent]

*August, 1976
Report #58
NAS-9-15000*

I. INTRODUCTION :

This program reads multispectral scanner data from a Universal format tape and outputs an intermediate data set in card image format for use as an input data set in various data analysis development programs. The general capabilities are summarized as follows:

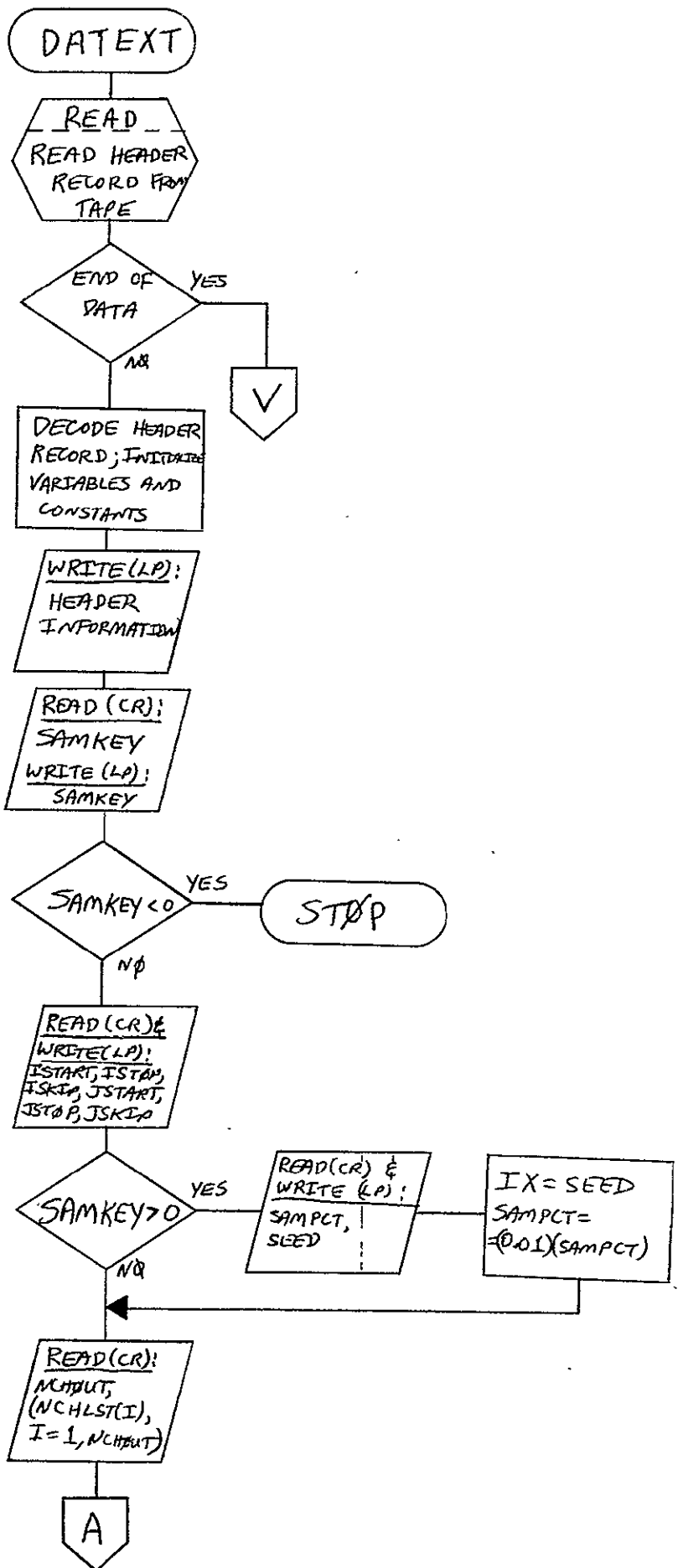
- 1) decode the header record of the universal format tape.
- 2) extract all or part of the channels on the universal format tape.
[The channel numbers are relative].
- 3) extract a rectangular region defined by first line [ISTART], last line [ISTOP], and a line skip factor [ISKIP] and analogous column or pixel values JSTART, JSTOP, AND JSKIP. [ISKIP or JSKIP = 1, means no lines are skipped.]
- 4) extract and label any region defined by a non-rectangular field or fields which is a subregion of .
- 5) randomly select a percentage SAMPCT of the regions or , which were defined in 3 or 4.

II. INPUT PARAMETERS :

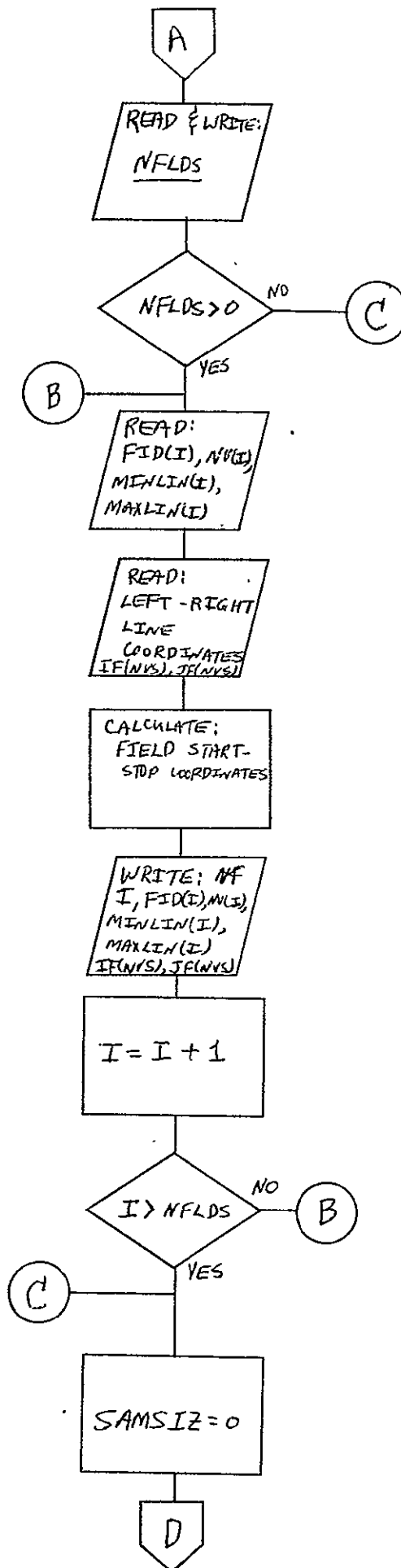
SAMKEY -1	-only header record is decoded
0	-deterministic sample is extracted
1	-random sample is extracted
SAMPCT	-if SAMKEY = 1, percent of data to be randomly sampled
SEED	-if SAMKEY = 1, initial seed for random number generator. (must be a positive odd integer)
ISTART	-beginning line for sample (absolute line number)
ISTOP	-last line for sample
ISKIP	-line skip factor (if ISKIP = 1, no lines are skipped)
JSTART	-beginning pixel for sample (relative pixel number)
JSTOP	-last pixel for sample
JSKIP	-pixel skip factor (if JSKIP = 1, no pixels are skipped)

NCHOUT	-number of channels to be output
NCHLST	-array of relative channel numbers of NCHOUT channels to be output
NFLDS	-number of non-rectangular fields to be defined (if NFLDS = 0, then the rectangular region defined by ISTART etc, is output)
FID	-array containing 8 character field ID for each field
NV	-array containing number of vertices for each non-rectangular field (if the field is a quadrilateral, then NV = 4)
MINLIN	-array containing the minimum line number for each field
MAXLIN	-array containing the maximum line number for each field
IF(J,I)	-two dimensional array containing the line coordinates of the Jth vertex of the Ith field for J = 1, . . . , NV+1 (the first coordinate is repeated as the NV+1 coordinate a la ERIPS)
JF(J,I)	-a two dimensional array containing the pixel coordinates of the Jth vertex of the Ith field for J = 1, . . . , NV+1 the first coordinate is repeated as the NV+1 coordinate a la ERIPS)

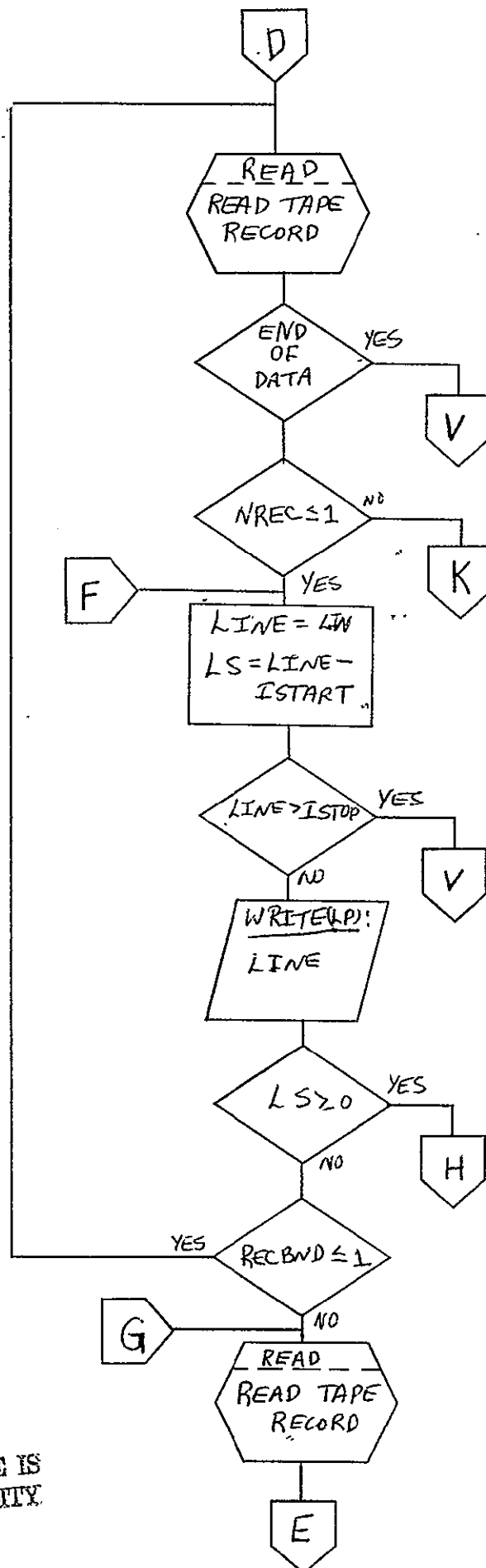
(the above vertices must be given in sequence such that the interior of the field lies to the right. See Appendix A for the ERIPS documentation for the FDLIN routine)



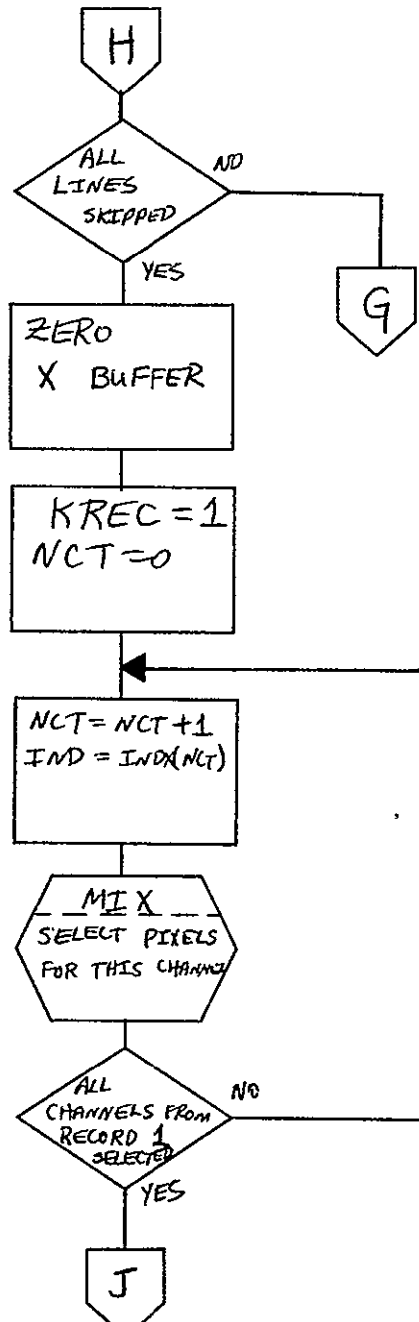
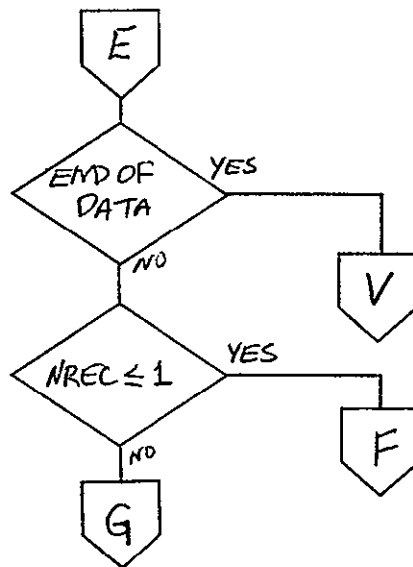
ORIGINAL PAGE IS
OF POOR QUALITY



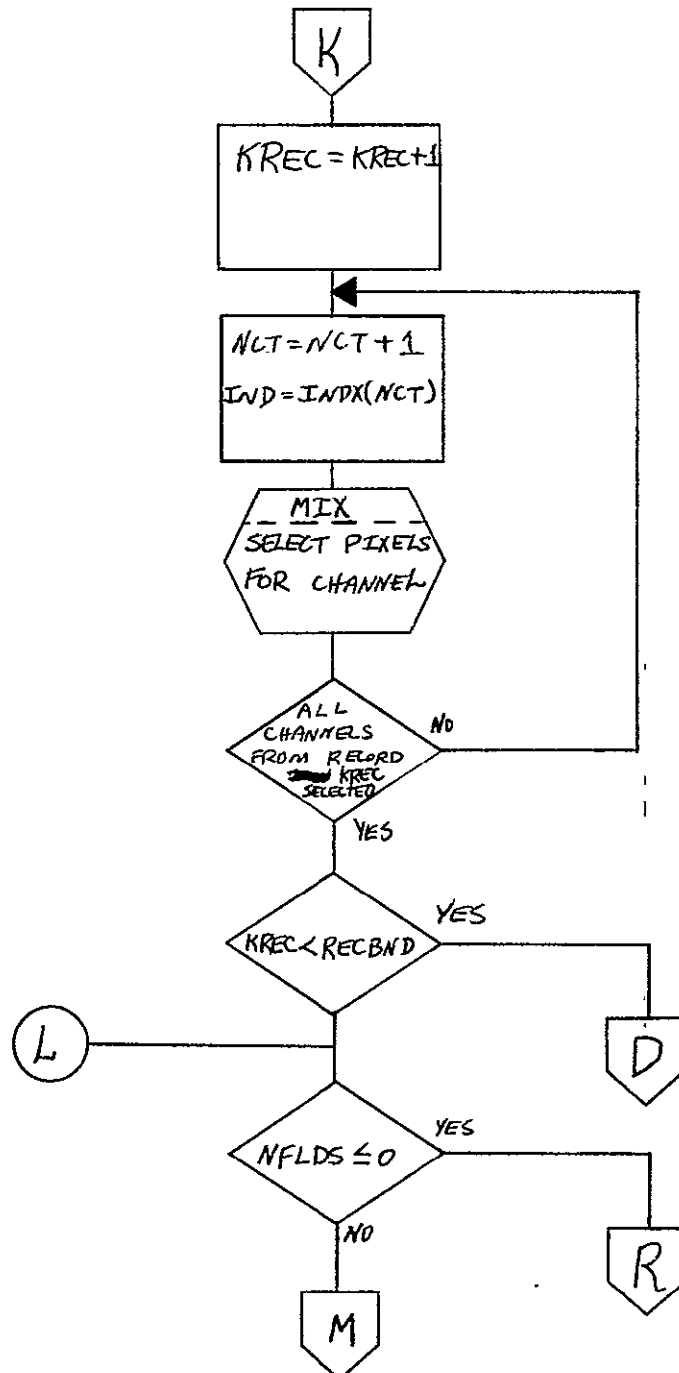
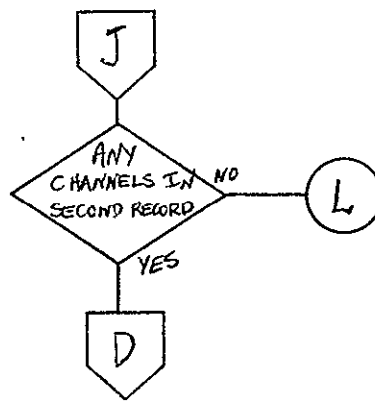
ORIGINAL PAGE IS
OF POOR QUALITY

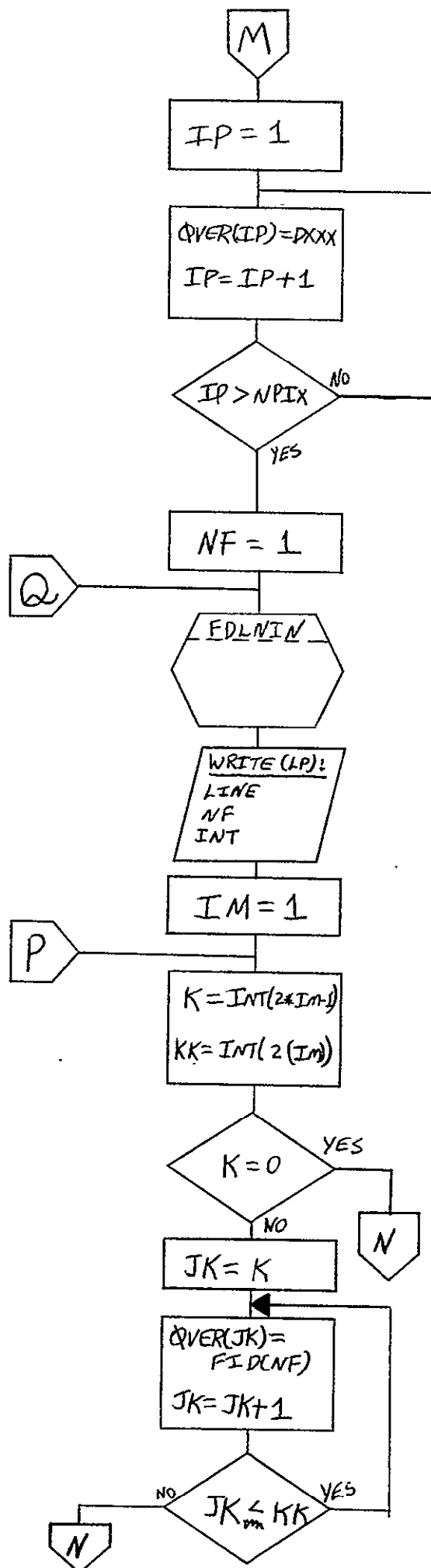


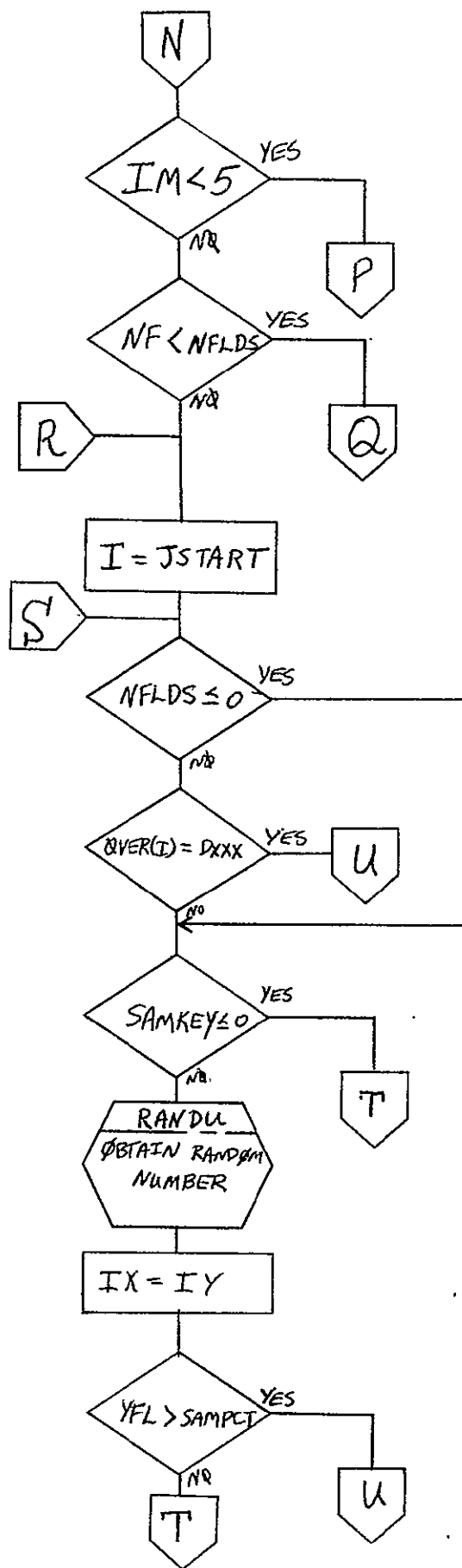
ORIGINAL PAGE IS
OF POOR QUALITY

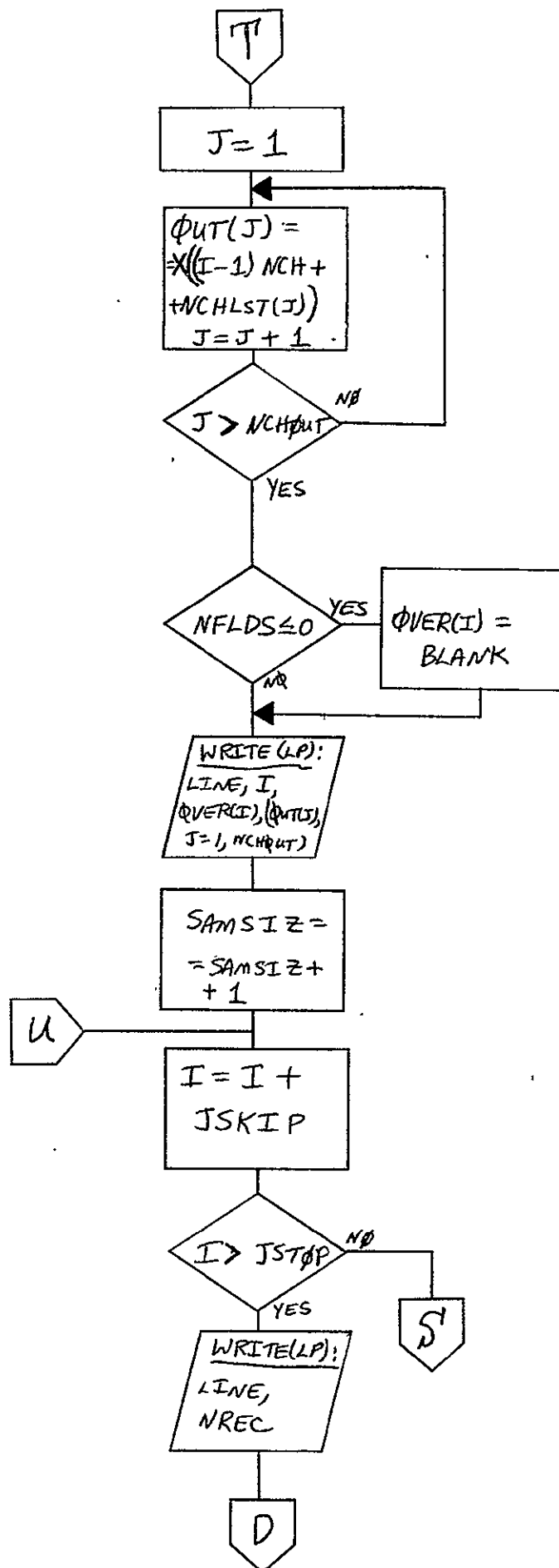


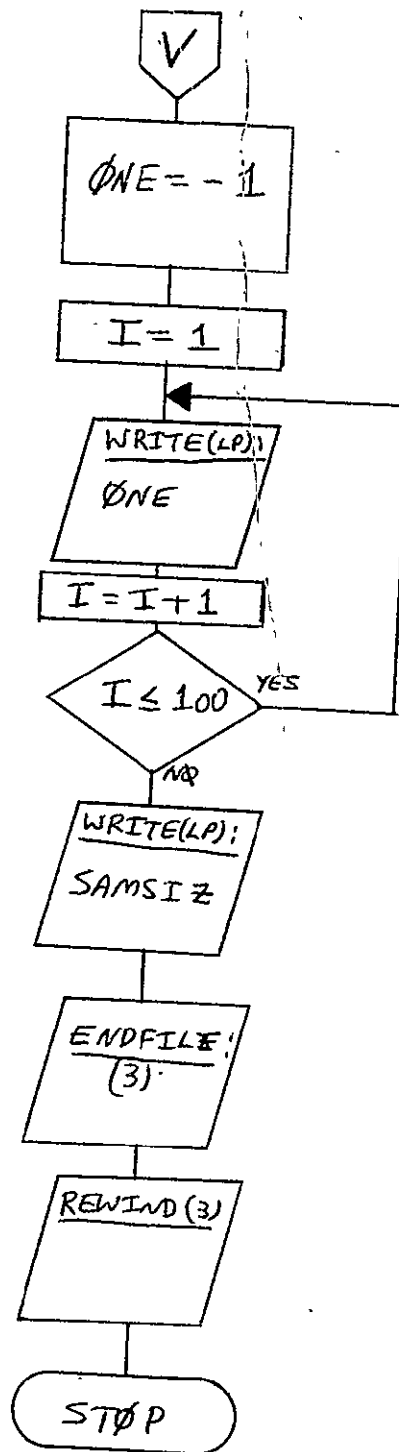
ORIGINAL PAGE IS
OF POOR QUALITY



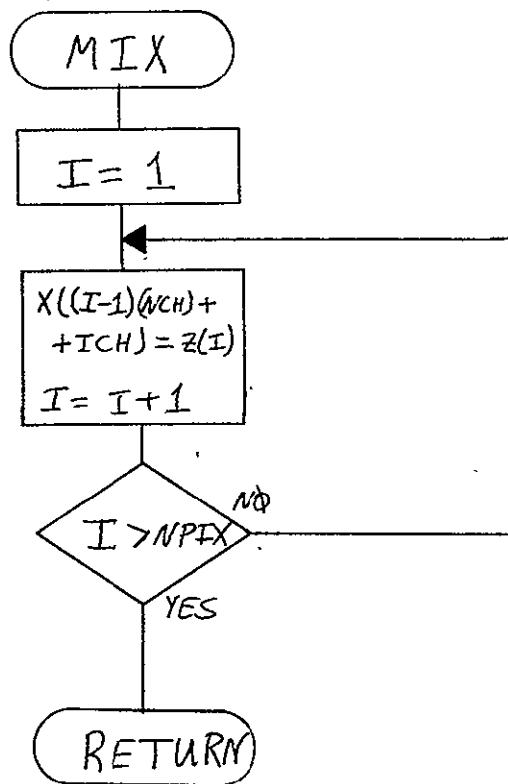




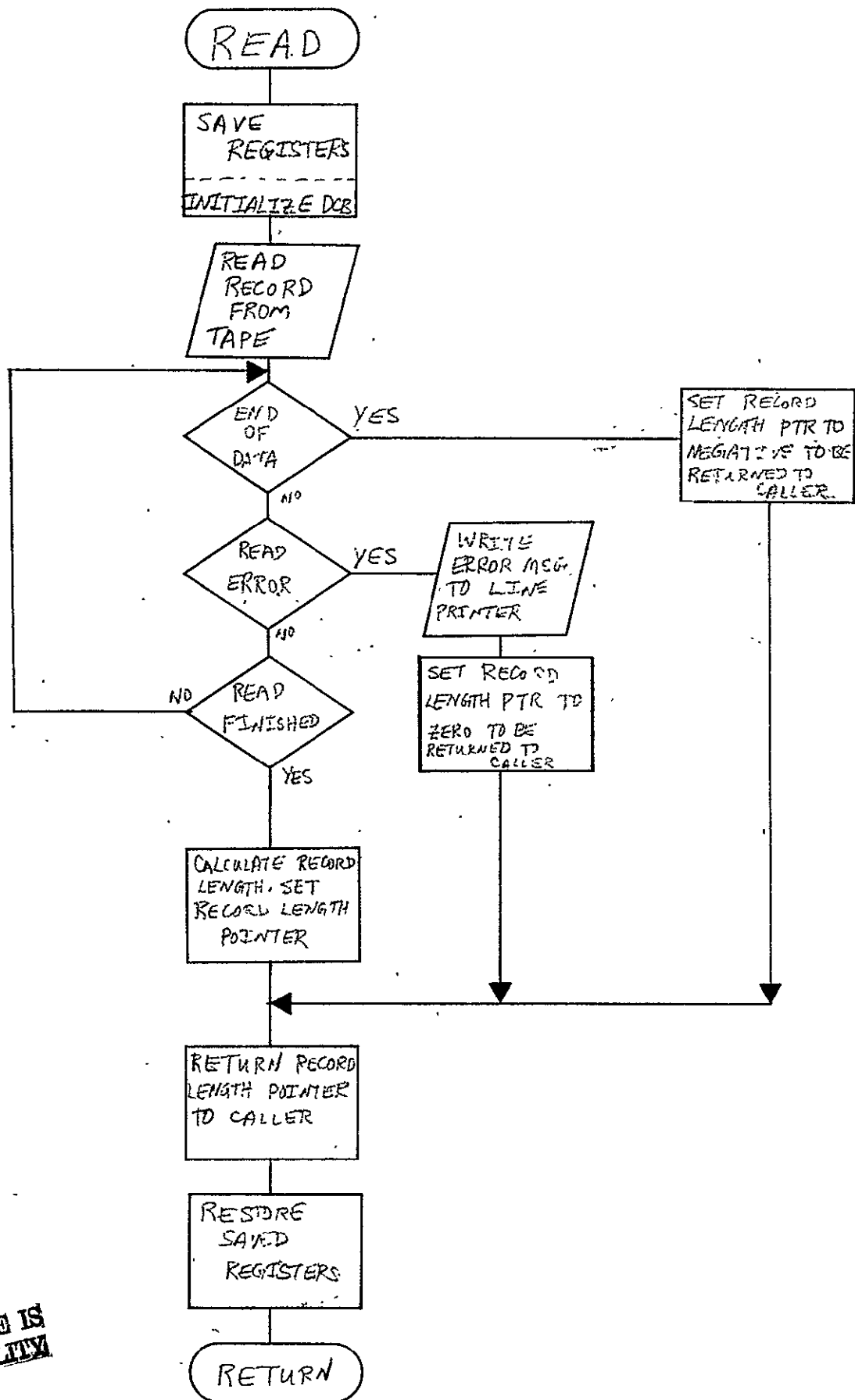




ORIGINAL PAGE IS
OF POOR QUALITY



ORIGINAL PAGE IS
OF POOR QUALITY



IV. INPUT FORMAT FOR PARAMETERS

REQ:	SAMKEY	(10X, I10)
	ISTART	
	ISTOP	
REQ:	ISKIP	(10X, I10)
	JSTART	
	JSTOP	
	JSKIP	
OPT:	SAMPCT	(10X, F10.0)
	SEED	(10X, I10)
REQ:	NCHOUT	(10X, I10)
	NCHLST	(10X, 16I2)
REQ:	NFLDS	(10X, I10)
	for I = 1, ..., NFLDS (if NFLDS = 0)	
	FID(I)	
OPT:	NV(I)	(A8, 2X, 3I5)
	MINLIN(I)	
	MAXLIN(I)	
	IF(J,I)	(11I5)
	JF(J,I)	(11I5)

V. FORMAT OF INPUT DATA SET

The Input Data Set is read from Fortran unit 1 (FT01F001) by the READ routine. The Input Data Set has the format of a Universal Format Image Data Tape described in NASA Earth Resources Data Format Control Book (TR-543).

VI. FORMAT OF OUTPUT DATA SET

For each NCH dimensional pixel $(X(I), I = 1, \dots, NCH)$ selected for output, the following record (80 bytes) is written onto Fortran unit 3 (FT03F001).

LINE number
PIXEL NUMBER
FID (if not applicable a blank is written)
X(NCHLST (1))
X(NCHLST (2))
.
.
.
X(NCHLST (NCHOUT))

The format is [2I4, A8, 16I4]. The logical record length is 80 bytes and the BLKSIZE is determined by the JCL card defining Fortran unit 3. (FT03F001).

VII. SUBROUTINES

MIX	-arranges data by pixel rather than by channel
RANDU	-random number generator (IBM SSP)
FOLNIN	-determines intersection of a non-rectangular files for a scan line. (Fortran version of PL1 ERIPS utility routine)
READ	-assembly language (360 OS) binary read routine (Hinman)

COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=50,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NCDECK,LOAD,MAP,NOEDIT,NOID,NOXREF

```

ISN 0002  INTEGER SEED
ISN 0003  INTEGER BEGVID,RECLNG,RECBND,ANCLNG,INDX(16),XXXX(2500),
          * ONE,SAMKEY,SAMSIZE,NCHLST(16)
ISN 0004  LOGICAL*1 Z(3060),Z2(2),X(10000),OUT(16)
ISN 0005  INTEGER*2 ZINT2,NREC,LIN,XX(5000)
ISN 0006  DOUBLE PRECISION OVER,BLANK,CXXX,FID
ISN 0007  DIMENSION FID(50),NV(50),MINLIN(50),MAXLIN(50),IF(12,50),
          * JF(12,50),INT(11),OVER(1000)
ISN 0008  DATA BLANK/' '
ISN 0009  DATA DXXX/'$$$$$$$/'
ISN 0010  DATA CLT/16*' ',SAMSIZE/0/,LIN/0/
ISN 0011  EQUIVALENCE (ZINT2,Z2(1)),(NREC,Z(1)),(LIN,Z(71)),
          * (X(1),XX(1)),(X(1),XXXX(1))

```

READ HEADER RECORD AND DECODE THE FOLLOWING VARIABLES

```

NCH  -- NUMBER OF CHANNELS
NCH1 -- NUMBER OF CHANNELS ON FIRST RECORD OF BAND
NCH2 -- NUMBER OF CHANNELS ON OTHER RECORDS OF BAND
RECLNG -- RECORD LENGTH
RECBND -- NUMBER OF RECORDS PER BAND
NPIX -- NUMBER OF PIXELS PER CHANNEL PER BAND
ANCLNG -- LENGTH OF ANCILLARY BLOCK ON FIRST RECORD OF BAND
BEGVID -- BEGIN VIDEO BYTE WITHIN SCAN
INDX  -- ARRAY OF INDICES FOR BEGINNING BYTE OF EACH CHANNEL
       WITHIN THE APPROPRIATE RECORD

```

```

ISN 0012  CALL READ(Z,LRC LG)
ISN 0013  IF (LRC LG.LT.0) GO TO 999
ISN 0015  ZINT2=0
ISN 0016  Z2(2)=Z(90)
ISN 0017  NCH=ZINT2
ISN 0018  Z2(1)=Z(92)
ISN 0019  Z2(2)=Z(93)
ISN 0020  BEGVID=ZINT2
ISN 0021  Z2(1)=Z(96)
ISN 0022  Z2(2)=Z(97)
ISN 0023  NPIX=ZINT2
ISN 0024  Z2(1)=Z(100)
ISN 0025  Z2(2)=Z(101)
ISN 0026  RECLNG=ZINT2
ISN 0027  ZINT2=0

```

ORIGINAL PAGE IS
OF POOR QUALITY

IIAXPFLI-ICAXPFLI

REFERENCES

1. Program Name - FDLNINT
2. Programmer - R. J. Decker
3. Language - PL/1
4. LINKEDIT Attributes - NCAL
5. Inputs - Scan Line Number
6. Outputs - Intercepts (pixel numbers) of scan line and field sides
7. Special Items - Calling sequence:

CALL FDLNINT(P,L);

where P = pointer to field definition table

L = 11 element vector declared

FIXED BIN (15)

L(11) should be loaded with the scan line number

On return, the L vector will contain the ordered pixel intercepts. (e.g., a return of

5	7	12	20	0	→	0
---	---	----	----	---	---	---

 indicates pixels 5 through 7 and pixels 12 through 20 are contained in the field.)

FUNCTIONAL DESCRIPTION

This subroutine will return the pixel numbers of those pixels on a given line that are contained within the boundaries of a field.

DETAILED LOGIC DESCRIPTION

IIAXPFLI examines the number of vertices of the input field to determine if the field is a line-field or a polygon. If the input field is a line-field, then the intercepts are determined as follows:

The intercept of the line-field and $L-0.5$ is calculated as $P = (X_2 - X_1) (L-0.5 - Y_1) / (Y_2 - Y_1) + X_1$. This calculation determines the projection of the intercept of the line-field and $L+0.5$ is calculated as $P = (X_2 - X_1) (L+0.5 - Y_1) / (Y_2 - Y_1) + X_1$. This calculation determines the projection of the intercept of $L+0.5$ onto L . These projections are examined to determine which is the left one (P_L) and which is the right one (P_R). P_L is set to the integral value of $P_L+0.5$ and P_R is set to the integral value of $P_R + 0.4999$.

Approval	Approval
<i>D.A. King 8/26/75</i>	

If the field is a polygon, then IIAXPFLI finds the pixel intercepts of a scan line and the sides of the input field.

There are three distinct cases and each is handled separately; (1) the scan line intersects a side but not at the endpoints (i.e., vertices), (2) the scan line intersects a vertex that is not an end of a horizontal line, and (3) the scan line is concurrent with a horizontal side of the field.

FUNCTIONAL FLOWCHART

See Figure 1.

IBM

Large Area Crop Inventory Experiment (LACIE)

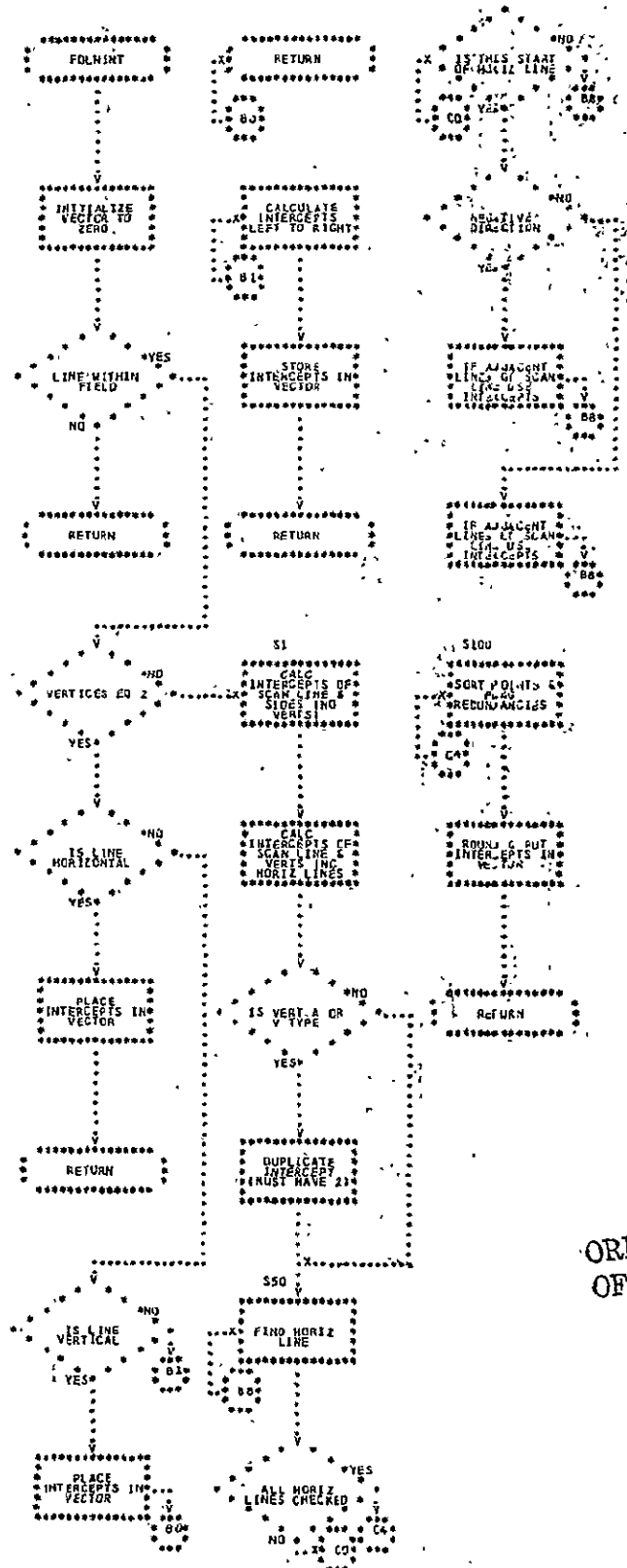
3.1 IAXPPLI-ICAXPPLI

Date 9/11/75

Rev

Page 3

Book: PROGRAM DOCUMENTATION



ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 1 FLOWCHART

```

ISN 0028      Z2(2)=Z(102)
ISN 0029      NCH2=ZINT2
ISN 0030      C      ZINT2=0
ISN 0031      Z2(2)=Z(104)
ISN 0032      RECBND=ZINT2
ISN 0033      C      Z2(1)=Z(105)
ISN 0034      Z2(2)=Z(106)
ISN 0035      ANCLNG=ZINT2
ISN 0036      C      Z2(1)=Z(1785)
ISN 0037      Z2(2)=Z(1786)
ISN 0038      NCH1=ZINT2
ISN 0039      C      ICT=0
ISN 0040      DO 20 I=1,NCH1
ISN 0041      ICT=ICT+1
ISN 0042      20    INDX(I)=ANCLNG+2+(I-1)*NPIX+1
ISN 0043      IF (RECBND.EQ.1) GO TO 40
ISN 0044      DO 30 I=2,RECBND
ISN 0045      DO 30 J=1,NCH2
ISN 0046      ICT=ICT+1
ISN 0047      INDX(ICT)=2+(J-1)*NPIX+1
ISN 0048      30    WRITE(6,200) NCH,NPIX,RECLNG,NCH1,NCH2,RECBND,ANCLNG,BEGVID
ISN 0049      40    WRITE(6,201) (I,INDX(I),I=1,NCH)
ISN 0050      WRITE(6,202) Z
ISN 0051      200  FORMAT(1H1,'NCH      = ',I6,/,
ISN 0052      *      'NPIX      = ',I6,/,
*      'RECLNG    = ',I6,/,
*      'NCH1      = ',I6,/,
*      'NCH2      = ',I6,/,
*      'RECBND    = ',I6,/,
*      'ANCLNG    = ',I6,/,
*      'BEGVID    = ',I6,/,
*      'INDX('',I2,'') = ',I8)
ISN 0053      201 FORMAT(1H1,'INDX('',I2,'') = ',I8)
ISN 0054      202 FORMAT(100(/,5(2X,I02Z)))

```

CCCCCCCCC

READ SAMPLING PARAMETERS

```

SAMKEY = -1 - ONLY HEADER RECORD IS DECODED
          0 - DETERMINISTIC SAMPLE
          1 - RANDOM SAMPLE
SAMPCT  - PERCENTAGE OF DATA TO BE SAMPLED RANDOMLY
SEED    - SEED FOR RANDOM NUMBER GENERATOR
ISTART  - BEGIN LINE FOR SAMPLE (ABSOLUTE LINE NUMBER)

```

ORIGINAL PAGE IS
OF 300 QUATERN

CCCCCCCC

ISTOP - LAST LINE FOR SAMPLE
 ISKIP - LINE SKIP FACTOR (IF ISKIP=0, NO LINES ARE SKIPPED)
 JSTART - BEGIN PIXEL FOR SAMPLE (RELATIVE PIXEL NUMBER)
 JSTOP - LAST PIXEL FOR SAMPLE
 JSKIP - PIXEL SKIP FACTOR (IF JSKIP=0, NO PIXELS ARE SKIPPED)
 NCHOUT - NUMBER OF CHANNELS TO BE OUTPUT
 NCHLST - ARRAY OF CHANNEL IDS TO BE OUTPUT (RELATIVE)

```

ISN 0055 READ(5,1000) SAMKEY
ISN 0056 WRITE(6,1007) SAMKEY
ISN 0057 IF(SAMKEY) 41,42,42
ISN 0058 41 STOP
ISN 0059 42 READ(5,1000) ISTART,ISTOP,ISKIP,JSTART,JSTOP,JSKIP
ISN 0060 WRITE(6,1008) ISTART,ISTOP,ISKIP,JSTART,JSTOP,JSKIP
ISN 0061 IF(SAMKEY) 44,44,43
ISN 0062 43 READ(5,1002) SAMPCT,SEED
ISN 0063 IX=SEED
ISN 0064 WRITE(6,1009) SAMPCT,SEED
ISN 0065 SAMPCT=SAMPCT/100.
ISN 0066 44 READ(5,1000) NCHOUT
ISN 0067 READ(5,1003) (NCHLST(I),I=1,NCHOUT)
ISN 0068 1000 FORMAT(10X,I10)
ISN 0069 1002 FORMAT(10X,F10.0,/,10X,I10)
ISN 0070 1003 FORMAT(10X,I10)
ISN 0071 1007 FORMAT(1H,'SAMKEY' = ',I10)
ISN 0072 1008 FORMAT(1H,'ISTART' = ',I10,/,
* 'ISTOP' = ',I10,/,
* 'ISKIP' = ',I10,/,
* 'JSTART' = ',I10,/,
* 'JSTOP' = ',I10,/,
* 'JSKIP' = ',I10)
ISN 0073 1009 FORMAT(' SAMPCT' = ',F10.2,/, 'SEED' = ',I10)
ISN 0074 1010 FORMAT(' NCHOUT' = ',I10)
ISN 0075 1011 FORMAT(' NCHLST' = ',16I5)
ISN 0076 READ(5,2000) NF LDS
ISN 0077 WRITE(6,2001) NF LDS
ISN 0078 2000 FORMAT(10X,I10)
ISN 0079 2001 FORMAT(1H,'NF LDS' = ',I10)
ISN 0080 IF(NF LDS) 440,440,438
ISN 0081 438 DO 439 NF=1,NF LDS
ISN 0082 READ(5,2002) FID(NF),NV(NF),MINLIN(NF),MAXLIN(NF)
ISN 0083 NVS=NV(NF) + 1
ISN 0084 READ(5,2003) (IF(J,NF),J=1,NVS)
ISN 0085 READ(5,2003) (JF(J,NF),J=1,NVS)
ISN 0086 DO 605 II=1,NVS
  
```

ORIGINAL PAGE IS
 OF POOR QUALITY

```

ISN 0087      J=NVS-11+1
ISN 0088      J1=J+1
ISN 0089      IF (J1,NF)=IF (J,NF)
ISN 0090      605 JF (J1,NF)=JF (J,NF)
ISN 0091      IF (1,NF)=IF (NVS,NF)
ISN 0092      JF (1,NF)=JF (NVS,NF)
ISN 0093      IF (NVS+2,NF)=IF (3,NF)
ISN 0094      JF (NVS+2,NF)=JF (3,NF)
ISN 0095      NV3=NVS+2
ISN 0096      WRITE (6,2004) NF
ISN 0097      WRITE (6,2005) FID(NF),NV(NF),MINLIN(NF),MAXLIN(NF)
ISN 0098      WRITE (6,2006) (JF(J,NF),J=1,NV3)
ISN 0099      439 WRITE (6,2007) (JF(J,NF),J=1,NV3)
ISN 0100      2002 FORMAT(A8,2X,3I5)
ISN 0101      2003 FORMAT(11I5)
ISN 0102      2004 FORMAT(5X,'FIELD = ',I10)
ISN 0103      2005 FORMAT(5X,'FIELD ID = ',I10,A8,' ',I10,'/',
      *          5X,'NV = ',I10,'/',
      *          5X,'MINLIN = ',I10,'/',
      *          5X,'MAXLIN = ',I10)
ISN 0104      2006 FORMAT(5X,'LINE = ',I12I5)
ISN 0105      2007 FORMAT(5X,'PIXEL = ',I12I5)
ISN 0106      440 CONTINUE

```

CCCC

WRITE DATA INTO CCB FORMAT

C

```

ISN 0107      SAMSIZ = 0
ISN 0108      50 CALL READ(Z,LRCLG)
ISN 0109      IF (LRCLG.LT.0) GO TO 999
ISN 0111      IF (NREC-1) 55,55,60

```

C

```

ISN 0112      55 LINE=LIN
ISN 0113      IF (LINE.GT.ISTOP) GO TO 999
ISN 0115      LS=LINE-ISTART
ISN 0116      WRITE (6,307) LINE
ISN 0117      307 FORMAT(20X,I10)
ISN 0118      IF (LS.GE.0) GO TO 552
ISN 0120      IF (RECBND.LE.1) GO TO 50

```

```

ISN 0122      550 CALL READ(Z,LRCLG)
ISN 0123      IF (LRCLG.LT.0) GO TO 999
ISN 0125      IF (NREC-1) 55,55,550

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

C
ISN 0126 552 LSM=LS/ISKIP*ISKIP-LS
ISN 0127 IF (LSM.NE.0) GO TO 550

C
ISN 0129 555 DO 56 I=1,2500
ISN 0130 56 XXXX(I)=0

C
ISN 0131 KREC=1
ISN 0132 NCT=0
ISN 0133 DO 57 I=1,NCH1
ISN 0134 NCT=NCT+1
ISN 0135 IND=INDX(NCT)
ISN 0136 57 CALL MIX(Z(IND),NCT,NPIX,X,NCH)
ISN 0137 IF (NCH2.EQ.0) GO TO 7329
ISN 0139 GO TO 50

C
ISN 0140 60 KREC=KREC+1
ISN 0141 DO 61 I=1,NCH2
ISN 0142 NCT=NCT+1
ISN 0143 IND=INDX(NCT)
ISN 0144 61 CALL MIX(Z(IND),NCT,NPIX,X,NCH)

C
ISN 0145 IF (KREC.LT.RECBND) GO TO 50

C
C WRITE DATA TO OUTPUT DATA SET
C
ISN 0147 7329 CONTINUE
ISN 0148 IF (NFLDS) 675,675,659

C
ISN 0149 659 DO 660 IP=1,NPIX
ISN 0150 660 OVER(IP)=DXXX
ISN 0151 DO 665 NF=1,NFLDS
ISN 0152 CALL FDLNIN(LINE,NV(NF),IF(1,NF),JF(1,NF),INT,MINLIN(NF),
* MAXLIN(NF))

C
C WRITE (6,6660) LINE,NF,INT
C 6660 FORMAT(30X,2I10,1I5)

C
ISN 0153 DO 668 IM=1,5
ISN 0154 K=INT(2*IM-1)
ISN 0155 KK=INT(2*IM)
ISN 0156 IF (K.EQ.0) GO TO 670
ISN 0158 DO 669 JK=K,KK
ISN 0159 669 OVER(JK)=FID(NF)
ISN 0160 670 CONTINUE
ISN 0161 668 CONTINUE
ISN 0162 665 CONTINUE

```

```

C
ISN 0163 675 CONTINUE
ISN 0164 DO 80 I=JSTART,JSTCP,JSKIP
C
ISN 0165 IF(NFLDS.LE.0) GO TO 680
C
ISN 0167 IF(OVER(I).EQ.DXXX) GO TO 80
C
ISN 0169 680 CONTINUE
ISN 0170 IF(SAMKEY) 75,75,70
ISN 0171 70 CALL RANDU(IX,IY,YFL)
ISN 0172 IX=IY
ISN 0173 IF(YFL.GT.SAMPCT) GO TO 80
C
ISN 0175 75 DO 78 J=1,NCHOUT
ISN 0176 78 OUT(J)=X(II=-1)*NCH + NCHLST(J))
C
ISN 0177 IF(NFLDS.LE.0) OVER(I)=BLANK
ISN 0179 WRITE(3,300) LINE,I,OVER(I),(OUT(J),J=1,NCHOUT)
ISN 0180 300 FORMAT(2I4,A8,16I4)
ISN 0181 SAMSI Z = SAMSI Z + 1
ISN 0182 80 CONTINUE
C
C
ISN 0183 WRITE(6,301) LINE,NREC
ISN 0184 301 FORMAT(2X,2I5)
ISN 0185 GO TO 50
C
ISN 0186 999 ONE=-1
ISN 0187 DO 90 I=1,100
ISN 0188 90 WRITE(3,400) ONE
ISN 0189 400 FORMAT(I4,76X)
ISN 0190 WRITE(6,405) SAMSI Z
ISN 0191 405 FORMAT(1X,SAMSI Z=*,I10)
ISN 0192 ENDFILE 3
ISN 0193 REWIND 3
ISN 0194 STOP
ISN 0195 END

```

COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=50,SIZE=0000K,
SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,NOID,NOXREF

```
ISN 0002 SUBROUTINE MIX(Z,ICH,NPIX,X,NCH)  
ISN 0003 LOGICAL*1 Z(1),X(1)  
ISN 0004 DO 1 I=1,NPIX  
ISN 0005 LOC=(I-1)*NCH+ICH  
ISN 0006 1 X(LOC)=Z(I)  
ISN 0007 RETURN  
ISN 0008 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

LEVEL 21.8 (JUN 74)

CS/360 FORTRAN H

DATE 7

COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=50,SIZE=0000K,
SOURCE,EBCCIC,NOLIST,NCDECK,LCAD,MAP,NOEDIT,NOID,NOXREF

```
ISN 0002 SUBROUTINE RANDU(IX,IY,YFL)
ISN 0003 IY=IX*65539
ISN 0004 IF(IY)5,6,6
ISN 0005 5 IY=IY+2147483647 + 1
ISN 0006 6 YFL=IY
ISN 0007 YFL=YFL*.4656613E-9
ISN 0008 RETURN
ISN 0009 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

COMPILER OPTIONS - NAME= MAIN,OPT=02,LINECNT=50,SIZE=0000K,
SOURCE,EBCCIC,NOLIST,NCDECK,LCAD,MAP,NOEDIT,NOID,NOXREF

```

----- ISN 0002 ----- SUBROUTINE FDLNIN (L,NV,Y,X,INT,MINLIN,MAXLIN)
ISN 0003 ----- INTEGER Y(12),X(12),INT(12),CUM
ISN 0004 ----- REAL PTS(10)
ISN 0005 ----- NV1=NV+1
ISN 0006 ----- DO 10 I=1,10
ISN 0007 ----- 10 INT(I)=0
ISN 0008 ----- IF (L.LT.MINLIN.GR.L.GT.MAXLIN) RETURN
ISN 0010 ----- 1 DO 15 I=1,10
ISN 0011 ----- 15 PTS(I)=0.
ISN 0012 ----- IPT=0
ISN 0013 ----- DO 12 I=2,NV1
ISN 0014 ----- IF (.NOT.((L.GT.MINO(Y(I),Y(I+1)).AND.L.LT.MAXO(Y(I),Y(I+1))))
----- * GO TO 12
ISN 0016 ----- IPT=IPT+1
ISN 0017 ----- PTS(IPT)=(FLOAT((L-Y(I))*(X(I+1)-X(I)))/
----- * (FLOAT(Y(I+1)-Y(I))+FLOAT(X(I)))
ISN 0018 ----- 12 CONTINUE
ISN 0019 ----- DO 14 I=2,NV1
ISN 0020 ----- IF (.NOT.((L.EQ.Y(I).AND.L.NE.Y(I-1).AND.L.NE.Y(I+1))) GO TO 14
ISN 0022 ----- IPT=IPT+1
ISN 0023 ----- PTS(IPT)=FLOAT(X(I))
ISN 0024 ----- IF (.NOT.((L.LT.Y(I-1).AND.L.LT.Y(I+1)).OR.((L.GT.Y(I-1).AND.
----- * L.GT.X(I+1)))) GO TO 14
ISN 0026 ----- IPT=IPT+1
ISN 0027 ----- PTS(IPT)=PTS(IPT-1)
ISN 0028 ----- 14 CONTINUE
ISN 0029 ----- J=1
ISN 0030 ----- 50 J=J+1
ISN 0031 ----- IF (J.GT.NV) GO TO 100
ISN 0033 ----- IF (Y(J).NE.L) GO TO 50
ISN 0035 ----- IF (Y(J+1).NE.U) GO TO 50
ISN 0037 ----- IF (X(J+1).LT.X(J)) GO TO 16
ISN 0039 ----- IF (Y(J-1).GE.L) GO TO 20
ISN 0041 ----- IPT=IPT+1
ISN 0042 ----- PTS(IPT)=X(J)
ISN 0043 ----- 20 IF (Y(J+2).GE.L) GO TO 21
ISN 0045 ----- IPT=IPT+1
ISN 0046 ----- PTS(IPT)=X(J+1)
ISN 0047 ----- 21 J=J+1
ISN 0048 ----- GO TO 50
ISN 0049 ----- 16 IF (Y(J-1).LE.L) GO TO 17
ISN 0051 ----- IPT=IPT+1
ISN 0052 ----- PTS(IPT)=X(J)
ISN 0053 ----- 17 IF (Y(J+2).LE.L) GO TO 18
ISN 0055 ----- IPT=IPT+1
ISN 0056 ----- PTS(IPT)=X(J+1)

```

```

ISN 0057      18      J=J+1
ISN 0058      GO TO 50
ISN 0059     -100-    CONTINUE
ISN 0060      IPT1=IPT-1
ISN 0061      DO 30 K=1,IPT1
ISN 0062      K1=K+1
ISN 0063      DO 30 I=K1,IPT
ISN 0064      IF (PTS(I).GE.PTS(K)) GO TO 30
ISN 0066      DUM=PTS(I)
ISN 0067      PTS(I)=PTS(K)
ISN 0068      PTS(K)=DUM
ISN 0069      30      CONTINUE
ISN 0070      IF (IPT.EQ.2) GO TO 103
ISN 0072      IPT2=IPT-2
ISN 0073      DO 40 I=2,IPT2
ISN 0074      IF (PTS(I).NE.PTS(I+1)) GO TO 40
ISN 0076      PTS(I)=-1
ISN 0077      PTS(I+1)=-1
ISN 0078      40      CONTINUE
ISN 0079      103     K=0
ISN 0080      DO 110 I=1,IPT,2
ISN 0081      IF (PTS(I).EQ.-1) GO TO 105
ISN 0083      K=K+1
ISN 0084      INT(K)=PTS(I)+.499
ISN 0085      105     CONTINUE
ISN 0086      IF (PTS(I+1).EQ.-1) GO TO 110
ISN 0088      K=K+1
ISN 0089      INT(K)=PTS(I+1)+.500
ISN 0090      110     CONTINUE
ISN 0091      120     IPT2=IPT-2
ISN 0092      DO 60 I=2,IPT2,2
ISN 0093      IF (INT(I).NE.INT(I+1)) GO TO 60
ISN 0095      INT(I)=0
ISN 0096      INT(I+1)=0
ISN 0097      60      CONTINUE
ISN 0098      IPT1=IPT-1
ISN 0099      DO 70 K=1,IPT1
ISN 0100      K1=K+1
ISN 0101      DO 65 I=K1,IPT
ISN 0102      IF (.NOT.(INT(I).NE.0.AND.INT(I).LT.INT(K).OR.INT(K).EQ.0)) GO TO 65
ISN 0104      DUM=INT(I)
ISN 0105      INT(I)=INT(K)
ISN 0106      INT(K)=DUM
ISN 0107      65      CONTINUE
ISN 0108      70      CONTINUE
ISN 0109      RETURN
ISN 0110      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

*READ

*
*
*

READ ERIPS LOG TAPE

*
*
*

LARRY HINMAN, EARTH RESOURCES PROGRAM OFFICE, PHILCO-FORD

*
*
*

CALL RDLOGT (BUFADR, RCLEN)

*
READ

CSECT

SAVE	(14,12),T,*	SAVE REGS
LR	2,15	SET BASE
LSING	READ,2	ASM BASE
LA	3,SAVE	NEW SAVE AREA ADDR
ST	3,8(13)	LSA
ST	13,4(3)	HSA
LR	13,3	SAVE AREA ADDR

*

L	3,0(1)	ADDR CF BUFFER
L	5,4(1)	ADDR OF WORD FOR RECORD LENGTH
LA	7,TAPEDCB	ADDR OF DCB
LSING	IHADCB,7	SECOND BASE
TM	DCBOFLGS,X'10'	TEST FOR OPEN
BO	INPUT	DCB IS OPEN

*

CPEN (TAPEDCB,,LPDCB,CUTPUT) INIT DCB'S

*

INPUT

DS	OH	READ RECCRDS FROM DCB
READ	INDECB,SF,TAPEDCB,(3),S'	READ RECORD

*

CHECK	INDECB	CHECK READ
-------	--------	------------

*

L	8,INDECB+16	IOB ADDR
LH	4,DCBBLKSI	RECCRD SIZE READ
SH	4,14(8)	LENGTH OF RECCRD READ

*

RTNO

DS	OH	SET RECORD LENGTH IN BYTES
ST	4,0(5)	RECCRD LENGTH TO CALLER

*

*

RETURN

DS	OH	RETURN LOGIC
L	13,SAVE+4	OLD SAVE AREA ADDR
RETURN	(14,12),T	RETURN TO CALLER

*

*

ENDDATA

DS	OH	END OF INPUT
MVI	0(5),X'FF'	SET RECCRD LENGTH TO NEGATIVE
B	RETURN	RETURN TO CALLER

*

*

ERROR

DS	OH	READ ERROR OCCURRED
ST	0,FIELD	DECE ADDR
UNPK	TMP(9),FIELD(5)	CONVERT TO PSEUDO-EBCCIC
TR	TMP(8),TABLE-240	CONVERT TO EBCCIC
MVC	ERRMSG+40(8),TMP	MOVE TO OUTPUT BUFFER

*

ST	1,FIELD	ERROR BITS AND DCB ADDR
UNPK	TMP(5),FIELD(5)	CONVERT TO PSEUDO-EBCCIC
TR	TMP(8),TABLE+240	CONVERT TO EBCCIC
MVC	ERRMSG+60(8),TMP	MOVE TO OUTPUT BUFFER

*

PLT	LPDCB,ERRMSG	OUTPUT ERROR MESSAGE
SR	4,4	ERROR OCCURRED
BR	14	RETURN TO SYSTEM

*****FT*LT*E*TS*

CA-TA

FIELD	DS	OF
TMP	DS	CL5
	DS	CL9

TAPEDCB - DCB--- MACRF=R, RECFM=U, BLKSIZE=8800, EODAD=ENDCAT-A,-----X
DSORG=PS, DDNAME=FT01FO01, SYNAD=ERRCR, DEVD=TA, EROPT=ACC

```

LPDCB      DCB      DSURG=PS,MACRF=PP,BLKSIZE=133,LRECL=133,RECFM=FBM,          X
                DDNAME=LP

```

ERRMSG DS OF
DC X'09',CL132 ***READ ERROR, RECCRD IGNORED***

TABLE	DS DC	OF C	0123456789ABCDEF
-------	----------	---------	------------------

..SAVE.....DS.....18F

CCBD DSORG=PS
~~END~~